

# Combinatoire élémentaire

Ce chapitre, premier d'une série de trois chapitres préliminaires, pose les définitions et les concepts de base d'algèbre et de combinatoire utilisés dans ce travail.

Dans le paragraphe 1.1, nous fixons nos notations et rappelons les définitions ainsi que les notions fondamentales sous-jacentes à cinq structures algébriques élémentaires : les classes combinatoires, les monoïdes, les ensembles partiellement ordonnés, les treillis et les espaces vectoriels combinatoires. Les mots sont des objets combinatoires à la fois très simples et très riches. Nous leur consacrons le paragraphe 1.2, où les permutations, qui sont des mots particuliers, sont également passées en revue. Enfin, dans le paragraphe 1.3, les définitions de base à propos des graphes et des arbres sont posées. Nous nous attardons plus particulièrement sur les arbres binaires, qui sont l'un des personnages principaux de ce mémoire.

## 1.1 Structures combinatoires et algébriques élémentaires

### 1.1.1 Classes combinatoires

**Définition 1.1.1.** Une classe combinatoire est un ensemble  $C$  muni d'une application taille  $|\cdot| \rightarrow \mathbb{N}$  telle que pour tout  $n \in \mathbb{N}$ , l'ensemble  $\{x \in C : |x| = n\}$  est fini.

Un objet combinatoire est un élément d'une classe combinatoire, et sa taille est son image par l'application  $|\cdot|$ . On note  $C_n$  l'ensemble des objets combinatoires de  $C$  de taille  $n$ .

À toute classe combinatoire  $C$  est associée sa série génératrice ordinaire  $\mathcal{S}_C(t)$ , qui est une série à coefficients entiers positifs, définie par

$$\mathcal{S}_C(t) := \sum_{n \geq 0} \#C_n t^n, \quad (1.1.1)$$

où la notation  $\#E$  désigne le cardinal de l'ensemble  $E$ . Dans certains cas, il est plus adapté de considérer la série génératrice exponentielle  $\mathcal{S}_C^e(t)$ , qui est définie par

$$\mathcal{S}_C^e(t) := \sum_{n \geq 0} \#C_n \frac{1}{n!} t^n. \quad (1.1.2)$$

Nous utilisons la notation  $[t^n]\mathcal{S}_C(t)$  pour accéder au coefficient de  $t^n$  dans la série  $\mathcal{S}_C(t)$ . Ainsi,  $[t^n]\mathcal{S}_C(t) = \#C_n$ .

Voici quelques exemples de classes combinatoires :

1. la classe combinatoire vide  $\emptyset$  ne possède aucun objet. Sa série génératrice vérifie  $\mathcal{S}_\emptyset(t) = 0$ .

2. La *classe combinatoire élémentaire*  $\mathcal{E}$  possède un unique objet de taille 0 que l'on note  $\epsilon$ . Sa série génératrice vérifie  $\mathcal{S}_{\mathcal{E}}(t) = 1$ .
3. La *classe combinatoire atomique*  $\mathcal{Z}$  possède un unique objet de taille 1 que l'on note  $\bullet$ . Sa série génératrice vérifie  $\mathcal{S}_{\mathcal{Z}}(t) = t$ .
4. Si  $(C, |\cdot|_C)$  et  $(D, |\cdot|_D)$  sont des classes combinatoires, l'*union* de  $C$  et de  $D$ , notée  $C + D$ , est la classe combinatoire qui contient les objets de  $C$  et de  $D$ , munie de l'application taille  $|\cdot|$  vérifiant  $|x| := |x|_C$  si  $x \in C$ , et  $|x| := |x|_D$  sinon.
5. De même, le *produit* de  $C$  et de  $D$ , noté  $C \times D$  est la classe combinatoire dont les objets sont des couples  $(c, d)$  où  $c \in C$  et  $d \in D$ , munie de l'application taille  $|\cdot|$  vérifiant  $|x| := |x|_C + |x|_D$ .
6. Il existe d'autres opérations bien connues sur les classes combinatoires qui à partir d'anciennes, en produisent de nouvelles. Par exemple, la construction *ensemble* prend en entrée une classe combinatoire  $C$  et produit la classe combinatoire dont les éléments sont des ensembles d'éléments de  $C$ . De même, les constructions *multi-ensemble* et *suite* produisent, sur l'entrée d'une classe combinatoire  $C$  respectivement la classe combinatoire dont les éléments sont des multi-ensembles d'éléments de  $C$  et la classe combinatoire dont les éléments sont des suites d'éléments de  $C$ . Ces constructions, ainsi que les séries génératrices correspondantes, sont décrites par exemple en [FS09].
7. L'ensemble des entiers naturels  $\mathbb{N}$  où la taille d'un objet est sa valeur est une classe combinatoire. De manière alternative,  $\mathbb{N}$  peut se définir comme étant la classe combinatoire obtenue par l'opération suite appliquée à  $\mathcal{Z}$ . Ses éléments de taille inférieure à cinq sont de la forme

$$\epsilon, \bullet, \bullet\bullet, \bullet\bullet\bullet, \bullet\bullet\bullet\bullet, \bullet\bullet\bullet\bullet\bullet,$$

et sa série génératrice est

$$\mathcal{S}_{\mathbb{N}}(t) = \frac{1}{1-t} = 1 + t + t^2 + t^3 + t^4 + \dots \quad (1.1.3)$$

8. La classe combinatoire des *mots* sur l'alphabet  $A := \{\mathbf{a}, \mathbf{b}\}$ , notée  $A^*$ , contient les suites finies d'éléments de  $A$  (voir le paragraphe 1.2.1). De manière alternative,  $A^*$  peut se définir comme étant la classe combinatoire obtenue par l'opération suite appliquée à  $A$  — ce dernier est vu comme une classe combinatoire qui possède exactement deux objets de taille 1. Ses éléments de taille inférieure à trois sont

$$\epsilon, \mathbf{a}, \mathbf{b}, \mathbf{aa}, \mathbf{ab}, \mathbf{ba}, \mathbf{bb}, \mathbf{aaa}, \mathbf{aab}, \mathbf{aba}, \mathbf{abb}, \mathbf{baa}, \mathbf{bab}, \mathbf{bba}, \mathbf{bbb},$$

et sa série génératrice est

$$\mathcal{S}(t) = \frac{1}{1-2t} = 1 + 2t + 4t^2 + 8t^3 + 16t^4 + 32t^5 + \dots \quad (1.1.4)$$

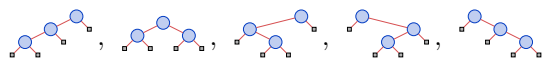
9. La classe combinatoire  $\mathcal{T}$  des *arbres binaires* (voir le paragraphe 1.3.3) est définie de manière récursive par

$$\mathcal{T} := \mathcal{E} + \mathcal{T} \times \mathcal{Z} \times \mathcal{T}. \quad (1.1.5)$$

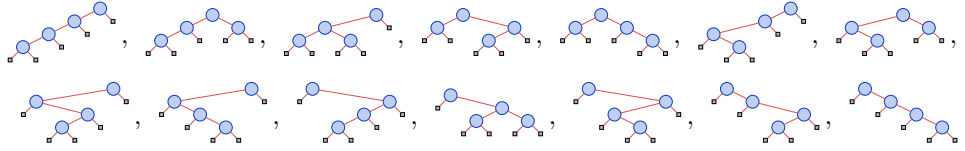
L'expression (1.1.5) spécifie le fait qu'un arbre binaire est soit vide, soit constitué d'un *nœud* et de deux autres arbres binaires. La taille d'un arbre binaire est son nombre de nœuds. Les arbres binaires de taille zéro, un et deux sont



ceux de taille trois,



et ceux de taille quatre,



### 1.1.2 Monoïdes

**Définition 1.1.2.** *Un monoïde est un ensemble  $M$  muni d'une loi de composition interne de la forme  $\cdot : M \times M \rightarrow M$  et telle que*

- (i)  $\cdot$  est associative, i.e.,  $(x \cdot y) \cdot z = x \cdot (y \cdot z)$  pour tous  $x, y, z \in M$  ;
- (ii)  $\cdot$  admet un élément neutre  $1 \in M$ , i.e.,  $x \cdot 1 = 1 \cdot x = x$  pour tout  $x \in M$ .

En fonction des besoins, nous noterons tantôt un monoïde sous la forme de triplet  $(M, \cdot, 1)$ , tantôt sous la forme de couple  $(M, \cdot)$ , ou même tout simplement en faisant référence à son ensemble sous-jacent  $M$  si le contexte le permet. Cette dernière convention de notation, où l'on fait référence à l'ensemble sous-jacent d'une structure algébrique pour parler de la structure algébrique elle-même, sera utilisée tout au long de ce mémoire.

Rappelons à présent quelques notions générales sur les monoïdes. Un monoïde  $(M, \cdot_M, 1_M)$  est *fini* si l'ensemble  $M$  est fini. La partie  $M' \subseteq M$  est un *sous-monoïde* de  $M$  si  $1_M \in M'$  et  $x \cdot_M y \in M'$  pour tous  $x, y \in M'$ . Soit  $G$  un sous-ensemble de  $M$ . Nous notons par  $G^*$  le *monoïde engendré* par  $G$ , c'est-à-dire le plus petit sous-monoïde de  $M$  qui contient  $G$ . Le monoïde  $M$  est *finiment engendré* s'il existe un ensemble fini  $G$  tel que  $M = G^*$ . Soit  $(N, \cdot_N, 1_N)$  un autre monoïde. Un *morphisme de monoïde* est une application  $\theta : M \rightarrow N$  telle que  $\theta(1_M) = 1_N$  et pour tous  $x, y \in M$ ,

$$\theta(x \cdot_M y) = \theta(x) \cdot_N \theta(y). \tag{1.1.6}$$

Les deux monoïdes  $M$  et  $N$  sont *isomorphes* s'il existe un morphisme de monoïde  $\theta : M \rightarrow N$  qui est également une bijection. Le monoïde  $M$  est *gradué* s'il existe une application  $\phi : M \rightarrow \mathbb{N}$  telle que  $\phi(x) = 0$  si et seulement si  $x = 1_M$ , et pour tous  $x, y \in M$ ,

$$\phi(x \cdot y) = \phi(x) + \phi(y). \tag{1.1.7}$$

En d'autres termes,  $M$  est gradué s'il admet un morphisme vers le monoïde  $(\mathbb{N}, +, 0)$  de telle sorte que le seul antécédent de 0 est  $1_M$ . Lorsqu'il existe une application  $\cdot^{-1} : M \rightarrow M$  telle que, pour tout  $x \in M$ ,

$$x \cdot_M x^{-1} = 1_M = x^{-1} \cdot_M x, \tag{1.1.8}$$

le monoïde  $M$  est un *groupe*. Dans ce cas, si  $x$  est un élément de  $M$ , l'élément  $x^{-1}$  est dit *l'inverse* de  $x$ .

Voici quelques exemples de monoïdes :

1. le *monoïde des entiers naturels*  $(\mathbb{N}, +, 0)$  est gradué par l'application identité, et finiment engendré par l'ensemble générateur  $\{1\}$ .
2. le *monoïde additif*  $(\mathbb{Z}, +, 0)$  est un groupe, l'inverse d'un élément étant son opposé. Le monoïde des entiers naturels est un sous-monoïde du monoïde additif.
3. Le *monoïde multiplicatif*  $(\mathbb{N}, \times, 1)$  est gradué par l'application qui associe à chaque élément le nombre de facteurs premiers qui le composent. De plus, ce monoïde n'est pas finiment engendré.
4. Le monoïde  $\max(\mathbb{N}, \max, 0)$  n'est ni gradué, ni finiment engendré.
5. Pour tout  $n \geq 1$ , le monoïde  $(\{E \subseteq [n]\}, \cap, [n])$ , où  $[n]$  désigne l'ensemble  $\{1, \dots, n\}$  est fini et n'est pas gradué.
6. Pour tout  $n \geq 1$ , le monoïde  $(\{E \subseteq [n]\}, \cup, \emptyset)$  est également fini et non gradué.

### 1.1.3 Ensembles partiellement ordonnés et treillis

Dans cette partie, nous rappelons les notions de base concernant les ensembles partiellement ordonnés et les treillis que nous utiliserons dans ce mémoire. Pour des notions plus précises et détaillées, le lecteur pourra consulter [BS81] ou [Sta11].

**Définition 1.1.3.** Un ensemble partiellement ordonné — ou poset — est un ensemble  $P$  muni d'une relation partielle  $\leq$  vérifiant les conditions

- (i) de réflexivité :  $x \leq x$  pour tout  $x \in P$  ;
- (ii) de transitivité :  $x \leq y$  et  $y \leq z$  impliquent  $x \leq z$  pour tous  $x, y, z \in P$  ;
- (iii) d'antisymétrie :  $x \leq y$  et  $y \leq x$  impliquent  $x = y$  pour tous  $x, y \in P$ .

Posons quelques notions de vocabulaire à propos des posets. Soit  $(P, \leq_P)$  un poset. Le poset  $P$  est fini si l'ensemble  $P$  est fini. Un ensemble  $P'$  est un sous-poset de  $P$  si  $P' \subseteq P$  et  $P'$  est muni de la relation d'ordre de  $P$  restreinte aux éléments de  $P'$ . La relation d'ordre  $\leq_P$  est totale si pour tous  $x, y \in P$ , nous avons  $x \leq_P y$  ou  $y \leq_P x$ . Soit  $(Q, \leq_Q)$  un autre poset. Si  $P$  et  $Q$  sont égaux en tant qu'ensembles et que pour tous  $x, y \in P$ ,

$$x \leq_P y \quad \text{implique} \quad x \leq_Q y, \quad (1.1.9)$$

alors  $Q$  est une extension de  $P$ . Si la relation d'ordre  $\leq_Q$  est de plus totale, alors  $Q$  est une extension linéaire de  $P$ . L'intervalle  $[x, z]$ , où  $x, z \in P$ , est l'ensemble

$$[x, z] := \{y \in P : x \leq_P y \leq_P z\}. \quad (1.1.10)$$

L'ensemble des couvertures de  $x \in P$  est l'ensemble

$$\{y \in P : \#[x, y] = 2\}. \quad (1.1.11)$$

Un sous-ensemble  $E$  de  $P$  est clos par intervalle si pour tous  $x, y \in E$ ,  $[x, y] \subseteq E$ . Une suite d'éléments  $x_1, x_2, \dots, x_\ell$  de  $P$  est une chaîne si

$$x_1 \leq_P x_2 \leq_P \dots \leq_P x_\ell ; \quad (1.1.12)$$

cette chaîne est de plus saturée si  $x_{i+1}$  couvre  $x_i$  pour tout  $i \in [\ell - 1]$ . Si l'élément  $x \in P$  vérifie pour tout  $y \in P$ ,

$$x \leq_P y \quad \text{implique} \quad x = y, \quad (1.1.13)$$

alors  $x$  est un élément maximal de  $P$ . Si l'élément  $y \in P$  vérifie pour tout  $x \in P$ ,  $x \leq_P y$ , alors  $y$  est le plus grand élément de  $P$ . On définit de la même manière la notion d'élément minimal et de plus petit élément en inversant la relation d'ordre  $\leq_P$ . Si  $P'$  est un sous-poset  $P$ , la borne supérieure de  $P'$ , notée  $\sup P'$ , est le plus petit élément, s'il existe, du sous-poset de  $P$  constitué des éléments

$$\{y \in P : \text{pour tout } x \in P', x \leq_P y\}. \quad (1.1.14)$$

De même, la borne inférieure de  $P'$ , notée  $\inf P'$ , est le plus grand élément, s'il existe, du sous-poset de  $P$

$$\{x \in P : \text{pour tout } y \in P', x \leq_P y\}. \quad (1.1.15)$$

S'il existe une application  $\phi : P \rightarrow \mathbb{N}$  telle que  $\phi(x) = 0$  si  $x$  est un élément minimal de  $P$ , et  $\phi(y) = \phi(x) + 1$  si  $y$  couvre  $x$  dans  $P$ , alors  $P$  est gradué. Le diagramme de Hasse de  $P$  est le graphe orienté (voir le paragraphe 1.3.1) dont l'ensemble de sommets est  $P$  et dont les arcs correspondent aux couples  $(x, y)$  tel que  $y$  couvre  $x$  dans  $P$ . Pour alléger nos représentations graphiques, les arcs d'un diagramme de Hasse ne seront pas orientés ; ils le sont implicitement par le fait que l'élément couvert se trouve plus haut que l'élément couvrant.

Certains posets admettent une structure plus fine :

**Définition 1.1.4.** Un treillis  $(L, \leq)$  est un poset tel que pour tous  $x, y \in L$ , l'ensemble  $\{x, y\}$  admet une borne inférieure et une borne supérieure.

Une définition des treillis, alternative à la définition 1.1.4, propose de voir une telle structure comme un ensemble  $L$  muni de deux lois de composition internes  $\vee$  et  $\wedge$  qui vérifient les conditions

- (i) d'associativité, de commutativité et d'idempotence (*i.e.*,  $x \vee x = x$  et  $x \wedge x = x$  pour tout  $x \in L$ );
- (ii) d'absorption :  $x \vee (x \wedge y) = x \wedge (x \vee y)$  pour tous  $x, y \in L$ .

À partir de la donnée d'un triplet  $(L, \vee, \wedge)$ , nous pouvons construire sur  $L$  une relation d'ordre  $\leq$ , définie pour tous  $x, y \in L$  par

$$x \leq y \quad \text{si et seulement si} \quad x \vee y = y \quad \text{et} \quad x \wedge y = x. \tag{1.1.16}$$

Inversement, à partir d'un treillis  $(L, \leq)$ , nous obtenons la structure précédemment définie en posant pour tous  $x, y \in L$ ,

$$x \vee y := \sup\{x, y\} \quad \text{et} \quad x \wedge y := \inf\{x, y\}. \tag{1.1.17}$$

Voici quelques exemples de posets et de treillis :

1. l'ensemble des entiers naturels  $\mathbb{N}$ , muni de la relation d'ordre usuelle  $\leq$ , est un poset. Il possède un plus petit élément, 0, mais pas de plus grand élément. Ce poset est de plus totalement ordonné et il s'agit ainsi d'un treillis, où les opérateurs  $\vee$  et  $\wedge$  correspondants vérifient  $x \vee y := \max\{x, y\}$  et  $x \wedge y := \min\{x, y\}$ . Son diagramme de Hasse est de la forme



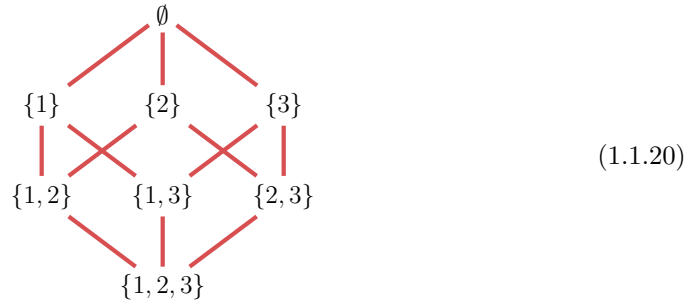
2. L'ensemble  $[n]$ , muni de la relation d'ordre  $|$  où l'on a  $x|y$  si  $x$  divise  $y$ , est pour tout  $n \geq 1$  un poset fini. Son plus petit élément est 1 et si  $n \geq 3$ , il ne possède pas de plus grand élément. En effet, les éléments  $n$  et  $n - 1$  sont maximaux et non comparables pour  $|$ . Le diagramme de Hasse pour le cas  $n := 8$  est



Lorsque l'on munit  $\mathbb{N}$  de la relation  $|$  on obtient un treillis où les opérateurs  $\vee$  et  $\wedge$  correspondants vérifient  $x \vee y := \text{ppcm}(x, y)$  et  $x \wedge y := \text{pgcd}(x, y)$ .

3. L'ensemble  $\{E \subseteq [n]\}$  admet pour tout  $n \geq 1$  une structure de treillis fini pour les opérateurs  $\vee$  et  $\wedge$  définis par  $E \vee F := E \cup F$  et  $E \wedge F := E \cap F$ . Ce treillis est connu sous

le nom de *treillis booléen* ou *hypercube*. La relation d'ordre correspondant à  $\vee$  et  $\wedge$  est la relation  $\subseteq$  d'inclusion ensembliste. Le diagramme de Hasse pour le cas  $n := 3$  est



### 1.1.4 Espaces vectoriels combinatoires

Pour le reste de ce mémoire, le symbole  $\mathbb{K}$  désigne un corps quelconque de caractéristique nulle. Sauf mention contraire, toutes les structures algébriques le nécessitant sont définies sur  $\mathbb{K}$ , et ceci implicitement.

Nous supposons connues du lecteur les notions basiques d'algèbre linéaire. Le but de ce paragraphe est d'introduire le vocabulaire que nous utiliserons dans le reste du texte, notamment dans le contexte des espaces vectoriels libres sur un ensemble. En effet, la plupart des espaces vectoriel que nous manipulerons sont des espaces vectoriels dont les bases sont indexées par des objets combinatoires.

Le *symbole de Kronecker*, noté  $\delta_{x,y}$ , est dans notre contexte à valeurs dans  $\mathbb{K}$ ; et  $\delta_{x,y} := 1_{\mathbb{K}}$  si  $x = y$ , et  $\delta_{x,y} := 0_{\mathbb{K}}$  sinon, où  $1_{\mathbb{K}}$  désigne l'élément neutre pour le produit de  $\mathbb{K}$  et  $0_{\mathbb{K}}$  — ou simplement 0 — désigne l'élément neutre pour l'addition.

Rappelons une construction qui à tout ensemble  $E$  associe un espace vectoriel. On appelle *support* d'une application  $f : E \rightarrow \mathbb{K}$  l'ensemble

$$\text{Supp}(f) := \{x \in E : f(x) \neq 0\}. \tag{1.1.21}$$

Soit

$$\text{Vect}(E) := \{f : E \rightarrow \mathbb{K} : f \text{ est à support fini}\}. \tag{1.1.22}$$

L'ensemble  $\text{Vect}(E)$  admet naturellement une structure d'espace vectoriel : si  $f, g \in \text{Vect}(E)$ , alors  $f + g$  est l'application  $x \mapsto f(x) + g(x)$ , et de même, si  $\lambda \in \mathbb{K}$ , alors  $\lambda f$  est l'application  $x \mapsto \lambda f(x)$ . Nous appelons  $\text{Vect}(E)$  *l'espace vectoriel libre* sur  $E$ . L'ensemble

$$\{\mathbf{F}_x := y \mapsto \delta_{x,y} : x \in E\} \tag{1.1.23}$$

forme une base évidente de  $\text{Vect}(E)$ . Ainsi, tout élément  $f$  de  $\text{Vect}(E)$  s'exprime sous la forme

$$f = \sum_{x \in \text{Supp}(f)} f(x) \mathbf{F}_x, \tag{1.1.24}$$

et nous pouvons voir  $\text{Vect}(E)$  comme l'espace vectoriel des sommes formelles finies d'éléments de  $E$  à coefficients dans  $\mathbb{K}$ . La base des  $\mathbf{F}$  est la *base fondamentale* de  $\text{Vect}(E)$ . Remarquons que dans certains cas, par souci de lisibilité, nous noterons simplement par  $x$  un élément  $\mathbf{F}_x$  de  $\text{Vect}(E)$ .

Nous rencontrerons dans ce mémoire des espaces vectoriels sous une forme particulière :

**Définition 1.1.5.** Soit  $C$  une classe combinatoire telle que  $\#C_0 = 1$ . L'espace vectoriel combinatoire engendré par  $C$  est l'espace vectoriel libre  $\text{Vect}(C)$ . Nous dirons que  $C$  est la classe combinatoire sous-jacente à  $\text{Vect}(C)$ .

Tout espace vectoriel combinatoire est *gradué*, c'est-à-dire qu'il se décompose comme une somme directe de la manière suivante :

$$\text{Vect}(C) = \bigoplus_{n \geq 0} \text{Vect}(C_n). \quad (1.1.25)$$

D'après la définition 1.1.1 des classes combinatoires, les espaces vectoriels  $\text{Vect}(C_n)$ , appelés *composantes homogènes de degré  $n$*  de  $\text{Vect}(C)$ , sont de dimension finie. Si  $V$  est un espace vectoriel combinatoire, nous emploierons la notation  $V^{(n)}$  pour faire référence à sa composante homogène de degré  $n$ .

La *série de Hilbert* d'un espace vectoriel combinatoire  $V$  est la série

$$F_V(t) := \sum_{n \geq 0} \dim V^{(n)} t^n, \quad (1.1.26)$$

qui n'est rien d'autre que la série génératrice de la classe combinatoire sous-jacente à  $V$ .

Soient maintenant  $V$  et  $W$  deux espaces vectoriels combinatoires. On rappelle qu'une *application linéaire*  $\phi : V \rightarrow W$  est une application qui vérifie pour tous  $\lambda \in \mathbb{K}$ , et  $x, y \in V$ ,

$$\phi(\lambda x) = \lambda \phi(x) \quad \text{et} \quad \phi(x + y) = \phi(x) + \phi(y). \quad (1.1.27)$$

Une *forme linéaire* sur  $V$  est une application linéaire  $\phi : V \rightarrow \mathbb{K}$  où  $\mathbb{K}$  est vu comme un espace vectoriel.

Soient  $V$  et  $W$  deux espaces vectoriels combinatoires engendrés respectivement par les classes combinatoires  $C$  et  $D$ . Le *produit tensoriel* de  $V$  avec  $W$ , noté  $V \otimes W$ , est l'espace vectoriel combinatoire tel que si  $\{E_c : c \in C\}$  est une base de  $V$  et  $\{F_d : d \in D\}$  est une base de  $W$ , alors

$$\{E_c \otimes F_d : (c, d) \in C \times D\} \quad (1.1.28)$$

est une base de  $V \otimes W$ . Ainsi, les éléments de cet espace vectoriel s'écrivent sur cette base comme combinaisons linéaires d'éléments de la forme  $E_c \otimes F_d$ . On appelle *tenseurs* de tels éléments. L'espace vectoriel  $V \otimes W$  est en outre muni, pour tous  $\lambda \in \mathbb{K}$ ,  $x, y \in V$ , et  $z, t \in W$ , des relations suivantes :

$$x \otimes (z + t) = x \otimes z + x \otimes t, \quad (1.1.29)$$

$$(x + y) \otimes z = x \otimes z + y \otimes z, \quad (1.1.30)$$

$$(\lambda x) \otimes z = x \otimes (\lambda z). \quad (1.1.31)$$

Soient  $V, V', W, W'$  des espaces vectoriels combinatoires et  $\phi : V \rightarrow W$  et  $\phi' : V' \rightarrow W'$  des applications linéaires. Le *produit tensoriel* de  $\phi$  et  $\phi'$ , noté  $\phi \otimes \phi'$ , est l'application linéaire

$$\phi \otimes \phi' : V \otimes V' \rightarrow W \otimes W' \quad (1.1.32)$$

définie pour tous  $x \in V$ ,  $y \in V'$ , par

$$\phi \otimes \phi'(x \otimes y) := \phi(x) \otimes \phi'(y). \quad (1.1.33)$$

Rappelons que l'*espace vectoriel dual* d'un espace vectoriel  $V$  de dimension finie est constitué de l'ensemble de ses formes linéaires. Le *dual gradué* — ou simplement *dual* — d'un espace vectoriel combinatoire  $V$  est la somme directe des espaces vectoriels duaux de chacune des composantes homogènes de  $V$ . Si  $\{F_c : c \in C\}$  est une base de  $V$ , nous noterons par  $\{F_c^* : c \in C\}$  sa *base duale*. Le *crochet de dualité* est la forme linéaire  $\langle \cdot, \cdot \rangle : V \otimes V^* \rightarrow \mathbb{K}$  définie par  $\langle F_c, F_d^* \rangle := \delta_{c,d}$ .

## 1.2 Combinatoire des mots et des permutations

Pour le reste de ce texte,  $A := \{a_1, a_2, a_3, \dots\}$  est un alphabet infini et totalement ordonné par  $a_i \leq a_j$  si et seulement si  $i \leq j$ .

### 1.2.1 Mots et opérations sur les mots

Un *mot* sur l'alphabet  $A$  est une suite finie de lettres de  $A$ . Sauf mention contraire, tous les mots que nous considérerons seront sur cet alphabet. La *taille*  $|u|$  d'un mot  $u$ , encore appelée sa *longueur*, est le nombre de lettres qui le constituent. L'unique mot de longueur 0 est le *mot vide* et est noté  $\epsilon$ . La notation  $u_i$  désigne la  $i^e$  lettre de  $u$ , pour tout  $i \in [|u|]$ . Soit  $u := u_1 u_2 \dots u_{|u|}$  un mot. Un *facteur* de  $u$  est un mot de la forme  $u_i u_{i+1} \dots u_j$  avec  $i \leq j \in [|u|]$ . Lorsque  $i = 1$ , ce facteur est également un *préfixe* de  $u$ , et lorsque  $j = |u|$ , il s'agit d'un *suffixe*. Un *sous-mot* de  $u$  est un mot de la forme  $u_{i_1} u_{i_2} \dots u_{i_k}$  où

$$1 \leq i_1 < i_2 < \dots < i_k \leq |u|. \quad (1.2.1)$$

On note par  $\text{Alph}(u)$  l'ensemble  $\{u_i : i \in [|u|]\}$ , qui est le plus petit alphabet sur lequel  $u$  est défini. Si  $E$  est un sous-ensemble de  $A$ , on note par  $u|_E$  la *restriction de  $u$  à l'alphabet  $E$* , c'est-à-dire le plus long sous-mot  $u'$  de  $u$  tel que  $\text{Alph}(u') = E$ . L'*évaluation* de  $u$  est le mot

$$\text{ev}(u) := a_1^{|u|_{a_1}} a_2^{|u|_{a_2}} \dots a_\ell^{|u|_{a_\ell}}, \quad (1.2.2)$$

où  $|u|_{a_i} := |u|_{\{a_i\}}$  dénote le *nombre d'occurrences* de la lettre  $a_i$  dans  $u$ . La plus grande (resp. plus petite) lettre de  $u$  est notée  $\max(u)$  (resp.  $\min(u)$ ). Si  $u$  admet un sous-mot de la forme  $u_i u_j$  avec  $u_i > u_j$ , le couple  $(i, j)$  est une *inversion* de  $u$ . De plus, si  $j = i + 1$ , l'indice  $i$  est qualifié de *descente*.

L'ensemble des mots sur  $A$  peut être muni de deux ordres totaux. Le premier, appelé *ordre lexicographique*, est noté  $\leq_{\text{lex}}$  et vérifie pour tous mots  $u$  et  $v$  sur  $A$ ,

$$u \leq_{\text{lex}} v \quad \text{si et seulement si} \quad \begin{array}{l} u \text{ est un préfixe de } v \\ \text{ou} \\ u = \mathbf{a}u' \text{ et } v = \mathbf{b}v' \text{ avec } \mathbf{a} < \mathbf{b}. \end{array} \quad (1.2.3)$$

Le second, appelé *ordre lexicographique par longueur*, est noté  $\leq_{\text{glex}}$  et vérifie pour tous mots  $u$  et  $v$  sur  $A$ ,

$$u \leq_{\text{glex}} v \quad \text{si et seulement si} \quad \begin{array}{l} |u| < |v| \\ \text{ou} \\ |u| = |v| \text{ et } u \leq_{\text{lex}} v. \end{array} \quad (1.2.4)$$

Une opération naturelle sur les mots, à savoir la *concaténation*, consiste à les placer bout à bout. Si  $u := u_1 \dots u_n$  et  $v := v_1 \dots v_m$  sont deux mots, la concaténation de  $u$  et  $v$ , notée  $u \cdot v$ , est le mot  $u_1 \dots u_n v_1 \dots v_m$ . Cette opération munit l'ensemble des mots d'une structure de monoïde, le *monoïde libre*, que l'on note  $A^*$ . Ce monoïde admet  $\epsilon$  comme élément neutre et est gradué par l'application qui à un mot associe sa longueur. Il est de plus engendré par l'ensemble  $A$ . Nous notons par  $A^+$  l'ensemble des mots non vides et par  $A^n$  l'ensemble des mots de longueur  $n$  sur  $A$ . De plus,  $\mathbb{K}\langle A \rangle$  désigne l'espace vectoriel engendré par les mots de  $A^*$ , ou, de manière équivalente, l'*espace vectoriel des polynômes non commutatifs* dont les variables sont des éléments de  $A$ . De manière générale, les notations  $E^*$ ,  $E^+$ ,  $E^n$  et  $\mathbb{K}\langle E \rangle$  sont valables pour un ensemble  $E$  quelconque, et désignent respectivement le monoïde libre des suites finies d'éléments de  $E$ , l'ensemble des suites finies et non vides d'éléments de  $E$ , l'ensemble des suites de longueur  $n$  d'éléments de  $E$  et l'espace vectoriel des polynômes non commutatifs sur l'ensemble  $E$  de variables.

Rappelons maintenant quelques opérations sur les mots. L'*image miroir* de  $u$  est le mot

$$u^\sim := u_{|u|} \dots u_2 u_1. \quad (1.2.5)$$



Soit  $v$  un mot. Le *produit de mélange*  $\sqcup$  de  $u$  et de  $v$  est défini linéairement sur  $\mathbb{K}\langle A \rangle$  par

$$u \sqcup v := \begin{cases} u & \text{si } v = \epsilon, \\ v & \text{si } u = \epsilon, \\ \mathbf{a}(u' \sqcup bv') + \mathbf{b}(au' \sqcup v') & \text{sinon, où } u = au' \text{ et } v = bv'. \end{cases} \quad (1.2.6)$$

Par exemple,

$$\begin{aligned} \mathbf{a}_1\mathbf{a}_2 \sqcup \mathbf{a}_2\mathbf{a}_1 &= \mathbf{a}_1\mathbf{a}_2\mathbf{a}_2\mathbf{a}_1 + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_2\mathbf{a}_1 + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_1\mathbf{a}_2\mathbf{a}_1 + \mathbf{a}_2\mathbf{a}_1\mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_1\mathbf{a}_1\mathbf{a}_2, \\ &= \mathbf{a}_1\mathbf{a}_2\mathbf{a}_1\mathbf{a}_2 + 2\mathbf{a}_1\mathbf{a}_2\mathbf{a}_2\mathbf{a}_1 + 2\mathbf{a}_2\mathbf{a}_1\mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_1\mathbf{a}_2\mathbf{a}_1. \end{aligned} \quad (1.2.7)$$

Soit  $\cdot^\#$  la transformation qui à un alphabet totalement ordonné  $B := \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \dots\}$  associe l'alphabet  $B^\# := \{\mathbf{b}_1^\#, \mathbf{b}_2^\#, \mathbf{b}_3^\#, \dots\}$  ordonné par  $\mathbf{b}_i^\# \leq \mathbf{b}_j^\#$  si et seulement si  $\mathbf{b}_j \leq \mathbf{b}_i$ . En imposant l'égalité  $\mathbf{b}_i^{\#\#} = \mathbf{b}_i$ , la transformation  $\cdot^\#$  est une involution sur les alphabets, *i.e.*,  $B^{\#\#} = B$ . L'*involution de Schützenberger*, également notée  $\cdot^\#$  est une involution sur  $A^*$  définie par

$$u^\# := u_{|u|}^\# \dots u_2^\# u_1^\#. \quad (1.2.8)$$

Par exemple,  $(\mathbf{a}_5\mathbf{a}_3\mathbf{a}_1\mathbf{a}_5\mathbf{a}_2)^\# = \mathbf{a}_2^\#\mathbf{a}_5^\#\mathbf{a}_1^\#\mathbf{a}_1^\#\mathbf{a}_3^\#\mathbf{a}_5^\#$ .

Dans un souci d'alléger les notations, les mots seront notés par des suites d'entiers, spécifiant sans ambiguïté les lettres de  $A$  correspondantes. Ainsi, à titre d'exemple, la suite 213211 sera la notation du mot  $\mathbf{a}_2\mathbf{a}_1\mathbf{a}_3\mathbf{a}_2\mathbf{a}_1\mathbf{a}_1$ .

## 1.2.2 Permutations, opérations sur les permutations et permutoèdre

Une *permutation* est une bijection  $\sigma : [n] \rightarrow [n]$ . Nous adopterons la *notation mot* des permutations;  $\sigma$  sera en effet notée par le mot  $\sigma(1)\sigma(2)\dots\sigma(n)$ . Les permutations sont donc vues comme des mots  $u$  qui vérifient  $\text{Alph}(u) = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{|u|}\}$ , et ainsi, les définitions du paragraphe précédent s'y appliquent. La *taille* d'une permutation est la longueur du mot qui la représente. La classe combinatoire des permutations est notée  $\mathfrak{S}$ . Si  $\sigma$  est de la forme  $\sigma = u\mathbf{a}_{i+1}v\mathbf{a}_i w$ , où  $\mathbf{a}_i, \mathbf{a}_{i+1} \in A$  et  $u, v$  et  $w$  sont des mots, alors  $\mathbf{a}_i$  est un *recul* de  $\sigma$ .

Chaque ensemble  $\mathfrak{S}_n$  est muni d'une relation d'ordre partielle  $\leq_p$  engendrée par la relation de couverture qui change un facteur de la forme  $\mathbf{ab}$  d'une permutation en le facteur  $\mathbf{ba}$  sous la condition que  $\mathbf{a} < \mathbf{b}$ . Autrement dit, une permutation  $\sigma$  est couverte par une permutation  $\nu$  si  $\sigma$  est de la forme  $\sigma = u\mathbf{ab}v$  et  $\nu$  de la forme  $\nu = u\mathbf{bav}$ . Cet poset est en fait un treillis, et est connu sous le nom de *permutoèdre droit* [GR63]. Nous l'appellerons dans ce qui suit simplement *permutoèdre*. La figure 1.1 montre le diagramme de Hasse du permutoèdre droit d'ordre 4.

Chaque ensemble  $\mathfrak{S}_n$  possède une structure de groupe où le produit de deux permutations est leur composition lorsque celles-ci sont vues comme des applications. L'inverse d'une permutation est son application inverse. Ce groupe est connu sous le nom de *groupe symétrique*. Nous avons ainsi par exemple  $35412 \cdot 41253 = 13524$  et  $35412^{-1} = 45132$ . Une permutation  $\sigma$  qui vérifie  $\sigma^{-1} = \sigma$  est une *involution*. Le groupe symétrique  $\mathfrak{S}_n$  admet la présentation suivante : c'est le groupe engendré par l'ensemble  $\{s_1, \dots, s_{n-1}\}$  soumis aux relations

$$s_i^2 = 1, \quad (1.2.9)$$

$$s_i \cdot s_j = s_j \cdot s_i \quad \text{pour tous } |i - j| \geq 2, \quad (1.2.10)$$

$$s_i \cdot s_{i+1} \cdot s_i = s_{i+1} \cdot s_i \cdot s_{i+1} \quad \text{pour tout } i \in [n - 2], \quad (1.2.11)$$

où 1 désigne l'élément neutre du groupe. Les éléments  $s_i$  sont appelés *transposition élémentaires* et agissent à droite sur les permutations par transposition de lettres. Nous avons ainsi

$$\sigma \cdot s_i = \sigma_1 \dots \sigma_{i+1} \sigma_i \dots \sigma_n. \quad (1.2.12)$$

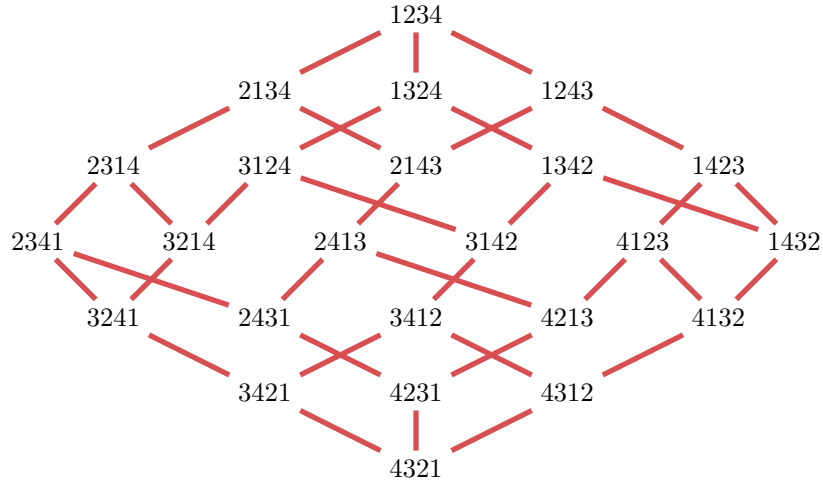


FIGURE 1.1 – Le diagramme de Hasse du permutoèdre droit d'ordre 4.

pour toute permutation  $\sigma \in \mathfrak{S}_n$  et  $i \in [n-1]$ .

Rappelons maintenant quelques opérations classiques sur les permutations. Soient  $\sigma \in \mathfrak{S}_n$  et  $\nu \in \mathfrak{S}_m$ . L'opération *over*  $/$  est définie par

$$\sigma / \nu := \sigma_1 \dots \sigma_n (\nu_1+n) \dots (\nu_m+n). \quad (1.2.13)$$

L'opération *under*  $\backslash$  est quant à elle définie par

$$\sigma \backslash \nu := (\nu_1+n) \dots (\nu_m+n) \sigma_1 \dots \sigma_n. \quad (1.2.14)$$

Par exemple,  $312 / 2314 = 3125647$  et  $312 \backslash 2314 = 5647312$ . La permutation  $\sigma$  est *connexe* si  $\sigma = \sigma^{(1)} / \sigma^{(2)}$  implique  $\sigma^{(1)} = \sigma$  ou  $\sigma^{(2)} = \sigma$ . De même, la permutation  $\sigma$  est *anti-connexe* si  $\sigma = \sigma^{(1)} \backslash \sigma^{(2)}$  implique  $\sigma^{(1)} = \sigma$  ou  $\sigma^{(2)} = \sigma$ . Par exemple, la permutation 31254 n'est pas connexe puisque  $31254 = 312 / 21$ , et la permutation 35421 n'est pas anti-connexe puisque  $35421 = 21 \backslash 132$ . En revanche, 35421 est connexe et 31254 est anti-connexe. Le *produit de mélange décalé*  $\sqcup$  est défini linéairement sur l'espace vectoriel combinatoire  $\text{Vect}(\mathfrak{S})$  par

$$\sigma \sqcup \nu := \sigma \sqcup ((\nu_1+|\sigma|) \dots (\nu_{|\nu|}+|\sigma|)). \quad (1.2.15)$$

Par exemple,

$$12 \sqcup 21 = 12 \sqcup 43 = 1243 + 1423 + 1432 + 4123 + 4132 + 4312. \quad (1.2.16)$$

La *standardisation*  $\text{std}$  est une application surjective de l'ensemble des mots vers l'ensemble des permutations qui associe à un mot  $u$  l'unique permutation  $\text{std}(u) \in \mathfrak{S}_{|u|}$  qui possède les mêmes inversions que  $u$ . Par exemple,  $\text{std}(314257423) = 416289735$ .





### 1.3 Combinatoire des graphes et des arbres

Nous avons besoin de définitions très basiques à propos des graphes. Après avoir rappelé celles-ci, nous introduisons le vocabulaire et autres définitions de base concernant les structures arborescentes qui vont intervenir dans ce mémoire.

### 1.3.1 Graphes


Un *graphe* est un couple  $(V, E)$  où  $V$  est un ensemble et  $E$  un ensemble de paires d'éléments de  $V$ . Les éléments de  $V$  sont appelés *sommets* et ceux de  $E$ , *arêtes*. Soit  $G := (V, E)$  un graphe. Un *chemin* est une suite de sommets de  $G$  de la forme  $(x_1, \dots, x_\ell)$  telle que  $\{x_i, x_{i+1}\} \in E$  pour tout  $i \in [\ell - 1]$ . Nous dirons que ce chemin *connecte* le sommet  $x_1$  au sommet  $x_\ell$  et que sa *longueur* est  $\ell$ . Ce chemin est aussi un *cycle* si  $\{x_\ell, x_1\} \in E$ . Le graphe  $G$  est *connexe* si pour tous sommets  $x, x' \in V$ , il existe au moins un chemin connectant  $x$  à  $x'$ . Tout sous-ensemble  $V'$  de  $V$  induit un *sous-graphe* de  $G$  : c'est le graphe dont les sommets sont ceux de  $V'$ , et les arêtes, celles de la forme  $\{x, y\}$  où  $\{x, y\} \in E$  et  $x, y \in V'$ . Une *composante connexe* de  $G$  est un sous-graphe connexe maximal de  $G$ . Le graphe  $G$  est *acyclique* s'il ne possède aucun cycle. Un *graphe étiqueté* est un graphe muni d'une application  $\tau : V \rightarrow A$  qui associe à chaque sommet  $x$  un élément de l'alphabet  $A$ .

Un graphe est *orienté* s'il est muni de deux applications  $\alpha : E \rightarrow V$  et  $\beta : E \rightarrow V$  telles que  $\alpha(\{x, y\}) = x$  si et seulement si  $\beta(\{x, y\}) = y$ . L'application  $\alpha$  associe à une arête son *origine*, tandis que l'application  $\beta$  lui associe son *but*. Une arête dans un graphe orienté est appelée *arc*. Un *chemin orienté* dans un graphe orienté est un chemin  $(x_1, \dots, x_\ell)$  tel que  $\beta(\{x_i, x_{i+1}\}) = \alpha(\{x_{i+1}, x_{i+2}\})$  pour tout  $i \in [\ell - 2]$ . Un sommet  $x$  d'un graphe orienté est une *source* si pour tout autre sommet  $y$ , il existe un chemin qui connecte  $x$  à  $y$ . Dans certains cas, il est plus approprié de voir un graphe orienté comme un couple  $(V, E)$  où  $E$  est un ensemble fini de sommets et  $V$  un ensemble de couples de sommets. Le couple  $(x, y) \in E$  indique en effet que le graphe contient une arête  $\{x, y\}$  telle que  $\alpha(\{x, y\}) = x$  et  $\beta(\{x, y\}) = y$ .

Dans nos représentations graphiques des graphes, les sommets seront dessinés par des cercles  et les arêtes par des segments  reliant deux sommets. Les arêtes d'un graphe orienté sont munies d'une flèche  qui pointe vers leur but. Un chemin arbitraire sera représentée par une ligne en zigzag  reliant les deux sommets qu'il connecte. L'étiquette d'un sommet d'un graphe étiqueté sera dessinée à l'intérieur du cercle qui le représente.

### 1.3.2 Structures arborescentes

Un *arbre* est un graphe à la fois connexe et acyclique. Il est d'usage courant d'appeler *nœuds* les sommets d'un arbre et *branches* ses chemins. Un *arbre enraciné* est un couple  $(G, r)$  où  $G$  est un arbre et  $r$  est un nœud de  $G$  appelé *racine*. Soit  $T$  un arbre enraciné de racine  $r$ , et  $x$  et  $y$  des nœuds de  $T$ . Si le chemin qui connecte  $r$  à  $y$  contient  $x$ , alors  $x$  est un *ancêtre* de  $y$ . Dans ce cas  $y$  est également un *descendant* de  $x$ . En particulier, la racine  $r$  est ancêtre de tous les nœuds de  $T$ . Le nœud  $x$  est le *parent* de  $y$  si  $x$  est un ancêtre de  $y$  et  $\{x, y\}$  est une arête de  $T$ . Dans ce cas,  $y$  est également un *fil* de  $x$ . Le *sous-arbre*  $T'$  de racine  $x$  dans  $T$  est l'arbre enraciné  $((V', E'), x)$  où  $V'$  est le sous-ensemble des nœuds de  $V$  qui sont descendants de  $x$ , et  $E'$  le sous-ensemble des arêtes  $\{y, z\}$  de  $E$  telles que  $y, z \in V'$ . Un *arbre plan* est un arbre enraciné muni d'une application  $\omega : V \rightarrow V^*$  telle que pour tous nœuds  $x$  et  $y$ , le mot  $\omega(x)$  est sans répétition de lettres, et  $\omega(x)$  admet une occurrence de  $y$  si et seulement si  $y$  est un fil de  $x$ . Ainsi,  $\omega$  impose un ordre total sur les fils des nœuds de l'arbre. Si  $x$  possède  $\ell$  fils,  $x$  est *d'arité*  $\ell$ , et le nœud  $\omega(x)_1$  est *le fils le plus à gauche* de  $x$ , et le nœud  $\omega(x)_\ell$  est *le fils le plus à droite* de  $x$ . Si  $T_1, T_2, \dots, T_\ell$  sont des arbres enracinés, la *greffe* de ces arbres, notée  $\wedge(T_1, T_2, \dots, T_\ell)$ , est l'arbre enraciné qui possède comme sous-arbres  $T_1, T_2, \dots, T_\ell$ , de gauche à droite.

Nous aurons besoin dans certains cas dans nos représentations graphiques de dessiner des sous-arbres arbitraires. Nous les schématiserons par des carrés . Dans le dessin d'un arbre enraciné, la racine est toujours le nœud dessiné le plus en haut. Les arbres plans seront dessinés de telle sorte que, de gauche à droite, les fils d'un nœuds  $x$  sont représentés conformément à l'ordre total imposé par le mot  $\omega(x)$ .

### 1.3.3 Arbres binaires et treillis de Tamari

Un *arbre binaire* est un arbre plan enraciné tel que chaque nœud possède 0 ou 2 fils. On distingue les nœuds sans fils de ceux qui en possèdent. Les nœuds sans fils seront en effet qualifiés de *feuilles* et les nœuds qui possèdent deux fils sont appelés *nœuds internes* ou simplement *nœuds*. Soit  $T$  un arbre binaire et  $x$  un nœud interne de  $T$ . Le fils le plus à gauche de  $x$  est le *fils gauche* de  $x$ , et le fils le plus à droite de  $x$  est le *fils droit* de  $x$ . De la même manière, on appelle *sous-arbre gauche* de  $x$  le sous-arbre de  $T$  ayant pour racine le fils gauche de  $x$ , et *sous-arbre droit* de  $x$  le sous-arbre de  $T$  ayant pour racine le fils droit de  $x$ . Par extension, le *sous-arbre gauche* de  $T$  est le sous-arbre gauche de sa racine, et le *sous-arbre droit* de  $T$  est le sous-arbre droit de sa racine. La *taille* d'un arbre binaire est son nombre de nœuds internes. La classe combinatoire des arbres binaires est notée  $\mathcal{T}$ . Le seul arbre binaire de taille 0 est composé d'une unique feuille, est appelé *arbre vide* et est noté  $\perp$ .

Une définition alternative des arbres binaires qui met en évidence leur structure récursive est la suivante : un arbre binaire est soit une feuille, soit un nœud qui possède un sous-arbre gauche et un sous-arbre droit auxquels sont attachés deux arbres binaires. De cette définition, on déduit que la série génératrice de  $\mathcal{T}$  vérifie l'équation

$$\mathcal{S}_{\mathcal{T}}(t) = 1 + t\mathcal{S}_{\mathcal{T}}(t)^2, \quad (1.3.1)$$

de laquelle on déduit

$$\mathcal{S}_{\mathcal{T}}(t) = \frac{1 - \sqrt{1 - 4t}}{2t} = \sum_{n \geq 0} \frac{1}{n+1} \binom{2n}{n} t^n. \quad (1.3.2)$$

Les premiers cardinaux des ensembles  $\mathcal{T}_n$  sont

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, \quad (1.3.3)$$

et constituent la célèbre suite de Catalan (suite [A000108](#) de [Slo]).

Dans nos représentations graphiques des arbres binaires, les feuilles seront dessinés par des petits carrés  $\square$ . Nous seront amenés, pour alléger la notation, à dessiner des arbres binaires sans leurs feuilles. Ceci ne porte aucun préjudice puisqu'il est possible de munir à nouveau un tel arbre binaire de feuilles d'une unique manière. La figure 1.2 montre un arbre binaire typique.

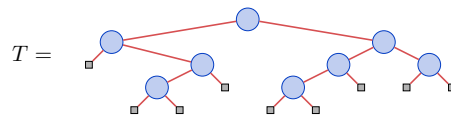


FIGURE 1.2 – Un arbre binaire de taille 8, dessiné avec ses feuilles.

Le *parcours infixe* d'un arbre binaire  $T \in \mathcal{T}_n$  non vide est la suite de ses nœuds obtenue récursivement en considérant le parcours infixe de son sous-arbre gauche, sa racine, puis le parcours infixe de son sous-arbre droit. Le nœud  $x$  de  $T$  est à *gauche* (resp. à *droite*) du nœud  $y$  de  $T$ , et on note  $x \rightsquigarrow_T y$  (resp.  $y \rightsquigarrow_T x$ ), si  $x$  apparaît avant (resp. après)  $y$  dans le parcours infixe de  $T$ . De plus, nous disons que le sous-arbre  $S$  de  $T$  est à *gauche* (resp. à *droite*) du nœud  $y$  de  $T$ , et on note  $S \rightsquigarrow y$  (resp.  $y \rightsquigarrow S$ ), si pour tout nœud  $x$  de  $S$ ,  $x \rightsquigarrow_T y$  (resp.  $y \rightsquigarrow_T x$ ). Si  $T$  est en plus étiqueté, la *lecture infixe* de  $T$  est le mot  $u_1 \dots u_n$  où pour tout  $i \in [n]$ ,  $u_i$  est l'étiquette du  $i^{\text{e}}$  nœud rencontré dans le parcours infixe de  $T$ .

La *canopée*  $\text{cnp}(T)$  d'un arbre binaire  $T$  est le mot sur l'alphabet  $\{0, 1\}$  obtenu en parcourant ses feuilles de gauche à droite, excepté la première et la dernière, en écrivant un 0 si la feuille considérée est orientée vers la droite, ou un 1 dans le cas contraire. Le terme « canopée » est

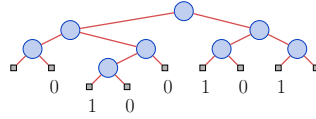


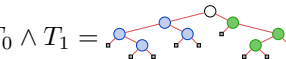


FIGURE 1.3 – Un arbre binaire dont la canopée est 0100101.

dû à Viennot [Vie04] mais la notion, elle, fut déjà utilisée en [LR98]. La figure 1.3 montre un exemple de canopée.

Rappelons quelques opérations sur les arbres binaires. Soient  $T_0$  et  $T_1$  deux arbres binaires. L'opérateur de greffe d'arbres binaires est noté dans ce contexte de manière infix, et ainsi,  $T_0 \wedge T_1$  est l'arbre binaire ayant  $T_0$  comme sous-arbre gauche et  $T_1$  comme sous-arbre droit. Si  $T_0$  et  $T_1$  sont étiquetés, on note  $T_0 \wedge_a T_1$  l'arbre binaire ayant  $T_0$  comme sous-arbre gauche et  $T_1$  comme sous-arbre droit, et où la racine est étiquetée par la lettre  $a$ . Les opérateurs *over*  $/$  et *under*  $\setminus$  de Loday et Ronco [LR02] se définissent comme suit. L'arbre binaire  $T_0 / T_1$  est obtenu en remplaçant la feuille la plus à gauche de  $T_1$  par la racine de  $T_0$ . Similairement,  $T_0 \setminus T_1$  est obtenu en remplaçant la feuille la plus à droite de  $T_0$  par la racine de  $T_1$ . Voici quelques exemples pour  $T_0 :=$   et  $T_1 :=$  .

$$T_0 \wedge T_1 = \text{}, \tag{1.3.4}$$

$$T_0 / T_1 = \text{}, \tag{1.3.5}$$

$$T_0 \setminus T_1 = \text{}. \tag{1.3.6}$$

Rappelons un autre opération fondamentale :

**Définition 1.3.1.** Soit  $T_0$  un arbre binaire et  $y$  un nœud de  $T_0$  ayant un sous-arbre gauche non vide. Soit  $S_0 := (A \wedge B) \wedge C$  le sous-arbre de racine  $y$  dans  $T_0$  et  $T_1$  l'arbre binaire obtenu en remplaçant  $S_0$  par  $A \wedge (B \wedge C)$  (voir la figure 1.4). Alors, la rotation droite en  $y$  envoie  $T_0$  sur  $T_1$ .

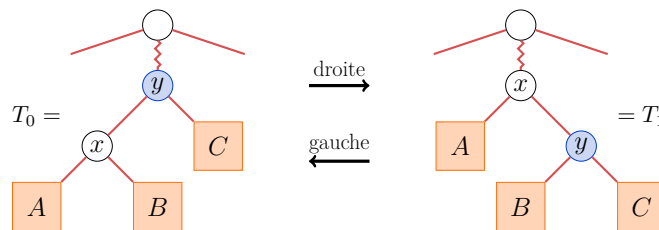


FIGURE 1.4 – L'opération de rotation dans un arbre binaire. La rotation droite en  $y$  envoie l'arbre binaire  $T_0$  sur l'arbre binaire  $T_1$ . La rotation gauche en  $y$  envoie quant à elle  $T_1$  sur  $T_0$ .

Si une rotation droite envoie un arbre binaire  $T_0$  sur un arbre binaire  $T_1$ , alors  $T_0$  est en relation partielle de Tamari avec  $T_1$ . Cette relation est notée  $T_0 \prec T_1$ . De plus, on notera  $T_0 \prec_y T_1$  si c'est une rotation droite en le nœud  $y$  de  $T_0$  qui envoie  $T_0$  sur  $T_1$ . Tant qu'il n'y a pas de confusion, le terme *rotation* signifiera *rotation droite* dans la suite.

On peut munir chaque ensemble  $\mathcal{T}_n$  d'une structure de treillis : le treillis de Tamari [Tam62]. Ce treillis peut être défini par l'intermédiaire d'objets combinatoires en bijection avec les arbres

binaires. La plupart des définitions portent sur des vecteurs d'entiers vérifiant certaines conditions [Sta99a], sur des forêts et des arbres binaires [Knu04] et sur des chemins de Dyck [BB09].

**Définition 1.3.2.** La relation d'ordre de Tamari  $\leq_{\mathbb{T}}$  est la clôture réflexive et transitive de la relation partielle de Tamari  $\prec$ . En d'autres termes, si  $T_0$  et  $T_\ell$  sont deux arbres binaires de même taille, nous avons  $T_0 \leq_{\mathbb{T}} T_\ell$  si  $\ell \geq 0$  et il existe une suite  $T_1, \dots, T_{\ell-1}$  d'arbres binaires telle que

$$T_0 \prec T_1 \prec \dots \prec T_{\ell-1} \prec T_\ell. \quad (1.3.7)$$

Pour tout  $n \geq 0$ , on note  $\mathbb{T}_n$  le poset  $(\mathcal{T}_n, \leq_{\mathbb{T}})$ .

La relation de Tamari est bien une relation d'ordre. En effet,  $\leq_{\mathbb{T}}$  est réflexive et transitive par définition. Pour montrer que  $\leq_{\mathbb{T}}$  est antisymétrique, considérons la statistique  $\phi : \mathcal{T} \rightarrow \mathbb{N}$  où  $\phi(T)$  est la somme, pour chaque nœud  $x$  de  $T$ , du nombre de nœuds dans le sous-arbre droit de  $x$ . Il est immédiat que si  $T_0 \prec T_1$  alors  $\phi(T_0) < \phi(T_1)$ , ce qui montre que  $\leq_{\mathbb{T}}$  est antisymétrique.

Huang et Tamari [HT72] ont montré que les posets  $\mathbb{T}_n$  forment un treillis. Ce sont les *treillis de Tamari*. La figure 1.5 montre les diagrammes de Hasse de  $\mathbb{T}_3$  et  $\mathbb{T}_4$ .

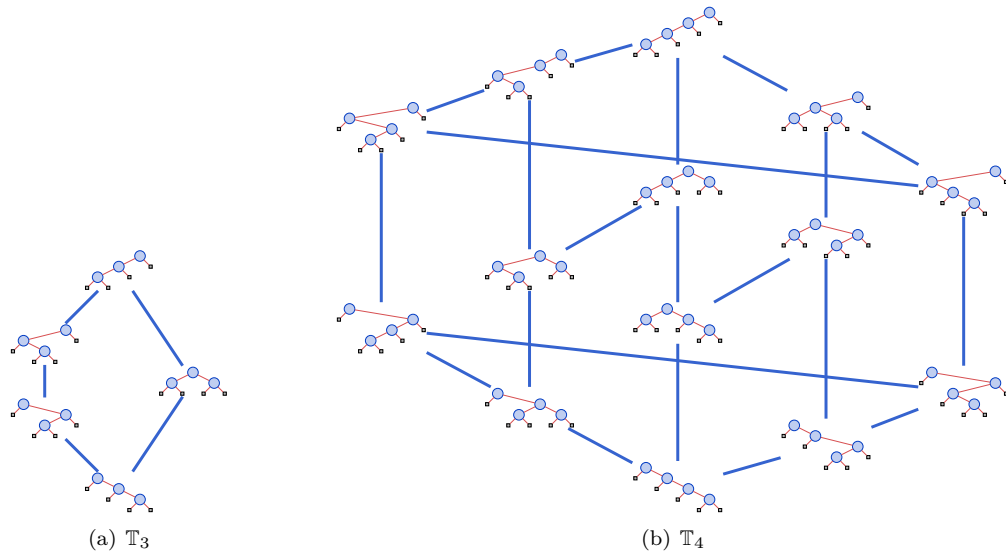


FIGURE 1.5 – Les diagrammes de Hasse des treillis de Tamari d'ordre 3 et 4.

Le plus petit élément du treillis de Tamari  $\mathbb{T}_n$  est le *peigne gauche* de taille  $n$ , c'est-à-dire l'arbre binaire à  $n$  nœuds dont chaque nœud possède un sous-arbre droit vide. De même, le plus grand élément de  $\mathbb{T}_n$  est le *peigne droit* de taille  $n$ , défini de façon similaire.

Soit  $T$  un arbre binaire étiqueté. Si pour tout nœud  $x$  de  $T$  étiqueté par une lettre  $b$ , pour tout nœud du sous-arbre gauche de  $x$  étiqueté par  $a$ , et tout nœud du sous-arbre droit de  $x$  étiqueté par  $c$ , nous avons l'inégalité  $a \leq b < c$ , alors  $T$  est un *arbre binaire de recherche droit*. Si nous avons à la place l'inégalité  $a < b \leq c$ , alors  $T$  est un *arbre binaire de recherche gauche* (voir la figure 1.6). Dans le cas où il n'y a pas de répétition dans les étiquettes de  $T$ , les notions d'arbres binaire de recherche droit et gauche coïncident.

Soit  $T$  un arbre binaire étiqueté. Si pour tous nœuds  $x$  et  $y$  de  $T$  étiquetés respectivement par  $a$  et  $b$ ,  $x$  est un ancêtre de  $y$  implique  $a \leq b$  (resp.  $b \leq a$ ), alors  $T$  est un *arbre binaire croissant* (resp. *décroissant*).

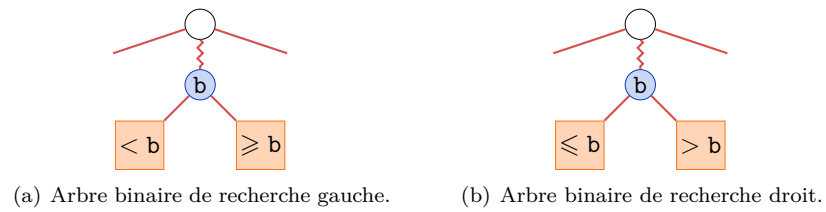


FIGURE 1.6 – Conditions sur les étiquettes des sous-arbres d'un arbre binaire de recherche.

La *forme* d'un arbre binaire étiqueté  $T$  désigne l'arbre binaire non étiqueté obtenu en oubliant les étiquettes de  $T$ .

